

Knowledge, Level of Symmetry, and Time of Leader Election *

Emanuele G. Fusco¹ and Andrzej Pelc^{†2}

¹Department of Computer, Control, and Management Engineering “Antonio Ruberti”
Sapienza, University of Rome. fusco@diag.uniroma1.it

¹Département d’informatique, Université du Québec en Outaouais, Gatineau, Québec J8X
3X7, Canada. pelc@uqo.ca

August 7, 2015

Abstract

We study the time needed for deterministic leader election in the *LOCAL* model, where in every round a node can exchange any messages with its neighbors and perform any local computations. The topology of the network is unknown and nodes are unlabeled, but ports at each node have arbitrary fixed labelings which, together with the topology of the network, can create asymmetries to be exploited in leader election. We consider two versions of the leader election problem: strong LE in which exactly one leader has to be elected, if this is possible, while all nodes must terminate declaring that leader election is impossible otherwise, and weak LE, which differs from strong LE in that no requirement on the behavior of nodes is imposed, if leader election is impossible. We show that the time of leader election depends on three parameters of the network: its diameter D , its size n , and its *level of symmetry* λ , which, when leader election is feasible, is the smallest depth at which some node has a unique view of the network. It also depends on the knowledge by the nodes, or lack of it, of parameters D and n .

keywords: leader election, anonymous network, level of symmetry

1 Introduction

1.1 The model and the problem

Leader election is one of the fundamental problems in distributed computing, first stated in [23]. Every node of a network has a boolean variable initialized to 0 and, after the election, exactly one node, called the *leader*, should change this value to 1. All other nodes should know which node is the leader. If nodes of the network have distinct labels, then leader election is always possible (e.g., the node with the largest label can become a leader). However, nodes may refrain from revealing their identities, e.g., for security or privacy reasons. Hence it is desirable to have leader election algorithms that do not rely on node identities but exploit asymmetries of the network due to its topology and to port labelings. With unlabeled nodes, leader election is impossible in symmetric networks [1].

A network is modeled as an undirected connected graph. We assume that nodes are unlabeled, but ports at each node have arbitrary fixed labelings $0, \dots, d - 1$, where d is the degree of the node. Throughout the paper, we will use the term “graph” to mean a graph with the above properties. We do not assume

*A preliminary version of this paper appeared in the Proc. of the 20th Annual European Symposium on Algorithms (ESA 2012), LNCS 7501, 479–490.

[†]Andrzej Pelc was partially supported by NSERC discovery grant and by the Research Chair in Distributed Computing at the Université du Québec en Outaouais.

any coherence between port labelings at various nodes. Nodes can read the port numbers. When sending a message through port p , a node adds this information to the message, and when receiving a message through port q , a node is aware that this message came through this port. The topology of the network is unknown to the nodes, but depending on the specification of the problem, nodes may know some numerical parameters of the network, such as the number n of nodes (size), and/or the diameter D . We consider two versions of the leader election problem (LE): *strong LE* and *weak LE*. In strong LE one leader has to be elected whenever this is possible, while all nodes must terminate declaring that leader election is impossible otherwise. Weak LE differs from strong LE in that no requirement on the behavior of nodes is imposed, if leader election is impossible. In both cases, upon election of the leader, every non-leader is required to know a path (coded as a sequence of ports) from it to the leader.

In this paper we investigate the time of leader election in the extensively studied *LOCAL* model [26]. In this model, communication proceeds in synchronous rounds and all nodes start simultaneously. In each round each node can exchange arbitrary messages with all its neighbors and perform arbitrary local computations. The time of completing a task is the number of rounds it takes. In particular, the time of weak LE is the number of rounds required by the last node to elect a leader if leader election is possible, and the time of strong LE is the number of rounds required by the last node to elect a leader if this is possible, and to terminate by declaring that this is impossible, otherwise.

It should be observed that the synchronous process of the *LOCAL* model can be simulated in an asynchronous network, by defining for each node separately its asynchronous round i in which it performs local computations, then sends messages stamped i to all neighbors, and waits until getting messages stamped i from all neighbors. (To make this work, every node is required to send a message with all consecutive stamps, until termination, possibly empty messages for some stamps.) All our results concerning time can be translated for asynchronous networks by replacing “time of completing a task” by “the maximum number of asynchronous rounds to complete it, taken over all nodes”.

If nodes have distinct labels, then time D in the *LOCAL* model is enough to solve any problem solvable on a given network, as after this time all nodes have an exact copy of the network. By contrast, in our scenario of unlabeled nodes, time D is often not enough, for example to elect a leader, even if this task is feasible. This is due to the fact that after time t each node learns only all paths of length t originating at it and coded as sequences of port numbers. This is far less information than having a picture of the radius t neighborhood. A node v may not know if two paths originating at it have the same other endpoint or not. It turns out that these ambiguities may force time much larger than D to accomplish leader election.

We show that the time of leader election depends on three parameters of the network: its diameter D , its size n , and its *level of symmetry* λ . The latter parameter is defined for any network (see Section 2 for the formal definition) and, if leader election is feasible, this is the smallest depth at which some node has a unique view of the network. The view at depth t from a node (formally defined in Section 2) is the tree of all paths of length t originating at this node, coded as sequences of port numbers on these paths. It is the maximal information a node can gain after t rounds of communication in the *LOCAL* model.

It turns out that the time of leader election also crucially depends on the *knowledge* of parameters n and/or D by the nodes. On the other hand, it does not depend on *knowing* λ , although it often depends on this parameter as well.

1.2 Our results

Optimal time of weak LE is shown to be $\Theta(D + \lambda)$, if either D or n is known to the nodes. More precisely, we give two algorithms, one working for the class of networks of given diameter D and the other for the class of networks of given size n , that elect a leader in time $O(D + \lambda)$ on networks with diameter D and level of symmetry λ , whenever election is possible. Moreover, we prove that this complexity cannot be improved. We show, for any values D and λ , a network of diameter D and level of symmetry λ on which leader election is possible but takes time at least $D + \lambda$, even when D , n and λ are known. If neither D nor n is known, then even weak LE is impossible [32].

For strong LE, we show that knowing only D is insufficient to perform it. Then we prove that, if only n is known, then optimal time is $\Theta(n)$. We give an algorithm working for the class of networks of given

		KNOWLEDGE			
		NONE	DIAMETER	SIZE	DIAMETER & SIZE
TASK	WLE	IMPOSSIBLE [32]	$\Theta(D + \lambda)$ (fast)		
	SLE	IMPOSSIBLE		$\Theta(n)$ (slow)	$\Theta(D + \lambda)$ (fast)

Table 1: Summary of the results

size n , which performs strong LE in time $O(n)$ and we show, for arbitrarily large n , a network G_n of size n , diameter $O(\log n)$ and level of symmetry 0, such that any algorithm performing correctly strong LE on all networks of size n must take time $\Omega(n)$ on G_n . Finally, if both n and D are known, then optimal time is $\Theta(D + \lambda)$. Here we give an algorithm, working for the class of networks of given size n and given diameter D which performs strong LE in time $O(D + \lambda)$ on networks with level of symmetry λ . In this case the matching lower bound carries over from our result for weak LE. Table 1 gives a summary of our results. The main difficulty of this study is to prove lower bounds on the time of leader election, showing that the complexity of the proposed algorithms cannot be improved for any of the considered scenarios.

The comparison of our results for various scenarios shows two exponential gaps. The first is between the time of strong and weak LE. When only the size n is known, strong LE takes time $\Omega(n)$ on some graphs of logarithmic diameter and level of symmetry 0, while weak LE is accomplished in time $O(\log n)$ for such graphs. The second exponential gap is for the time of strong LE, depending on the knowledge provided. While knowledge of the diameter alone does not help to accomplish strong LE, when this knowledge is added to the knowledge of the size, it may exponentially decrease the time of strong LE. Indeed, strong LE with the knowledge of n alone takes time $\Omega(n)$ on some graphs of logarithmic diameter and level of symmetry 0, while strong LE with the knowledge of n and D is accomplished in time $O(\log n)$, for such graphs.

1.3 Related work

Leader election was first studied for rings, under the assumption that all labels are distinct. A synchronous algorithm, based on comparisons of labels, and using $O(n \log n)$ messages was given in [18]. It was proved in [14] that this complexity is optimal for comparison-based algorithms. On the other hand, the authors showed an algorithm using a linear number of messages but requiring very large running time. An asynchronous algorithm using $O(n \log n)$ messages was given, e.g., in [27] and the optimality of this message complexity was shown in [6]. Deterministic leader election in radio networks has been studied, e.g., in [19, 20, 24] and randomized leader election, e.g., in [29].

Many authors [2, 3, 5, 11, 21, 22, 28, 30, 32] studied various computing problems in anonymous networks. In particular, [4, 32] characterize networks in which leader election can be achieved when nodes are anonymous. In [31] the authors study the problem of leader election in general networks, under the assumption that labels are not unique. They characterize networks in which this can be done and give an algorithm which performs election when it is feasible. They assume that the number of nodes of the network is known to all nodes. In [13] the authors study feasibility and message complexity of sorting and leader election in rings with nonunique labels, while in [12] the authors provide algorithms for the generalized leader election problem in rings with arbitrary labels, unknown (and arbitrary) size of the ring and for both synchronous and asynchronous communication. In [16] the leader election problem is approached in a model based on mobile agents. Characterizations of feasible instances for leader election and naming problems have been provided in [7, 8, 9]. Memory needed for leader election in unlabeled networks has been studied in [15]. To the best of our knowledge, the problem of time of leader election in arbitrary unlabeled networks has never been studied before.

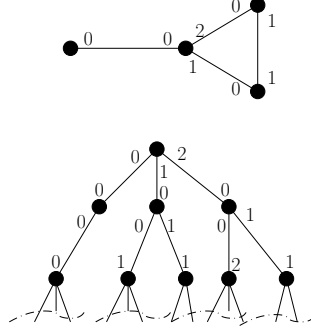


Figure 1: A graph and a fragment of the corresponding view from its degree 3 node.

2 Preliminaries

We say that leader election is *possible* for a given graph, or that this graph is a *solvable graph*, if there exists an algorithm which performs LE for this graph.

We consider two versions of the leader election task for a class \mathcal{C} of graphs :

- Weak LE. Let G be any graph in class \mathcal{C} . If leader election is possible for the graph G , then it is accomplished.
- Strong LE. Let G be any graph in class \mathcal{C} . If leader election is possible for the graph G , then it is accomplished. Otherwise, all nodes eventually declare that the graph is not solvable and stop.

Hence weak LE differs from strong LE in that, in the case of impossibility of leader election, no restriction on the behavior of nodes is imposed: they can, e.g., elect different leaders, or no leader at all, or the algorithm may never stop.

We will use the following notion from [32]. Let G be a graph and v a node of G .

Definition 2.1 (View) *The view from v is the infinite rooted tree $\mathcal{V}(v)$ with labeled ports, defined recursively as follows. $\mathcal{V}(v)$ has the root x_0 corresponding to v . For every node v_i , $i = 1, \dots, k$, adjacent to v , there is a neighbor x_i in $\mathcal{V}(v)$ such that the port number at v corresponding to edge $\{v, v_i\}$ is the same as the port number at x_0 corresponding to edge $\{x_0, x_i\}$, and the port number at v_i corresponding to edge $\{v, v_i\}$ is the same as the port number at x_i corresponding to edge $\{x_0, x_i\}$. Node x_i , for $i = 1, \dots, k$, is now the root of the view from v_i .*

See Fig. 1 for an example of a view. The following proposition directly follows from [32] and expresses the feasibility of leader election in terms of views.

Proposition 2.1 *Let G be a graph. The following conditions are equivalent:*

1. Leader election is possible in G ;
2. Views of all nodes are different;
3. There exists a node with a unique view.

In particular, the impossibility of deterministic leader election, when there are two nodes with identical views, is implied by proposition 4 from [32], while corollary 1 from [32] implies that one node has a unique view if and only if all views of the nodes are different.

By $\mathcal{V}^t(v)$ we denote the view $\mathcal{V}(v)$ truncated to depth t . We call it the view of v at depth t . In particular, $\mathcal{V}^0(v)$ consists of the node v , together with its degree. The following proposition was proved in [25].

Proposition 2.2 *For a n -node graph, $\mathcal{V}(u) = \mathcal{V}(v)$, if and only if $\mathcal{V}^{n-1}(u) = \mathcal{V}^{n-1}(v)$.*

For graphs of sublinear diameter a better bound than that of Proposition 2.2 has been proved by Hendrickx [17] after the publication of the conference version of this paper; for an n -node graph of diameter d , $\mathcal{V}(u) = \mathcal{V}(v)$, if and only if $\mathcal{V}^k(u) = \mathcal{V}^k(v)$, where $k \in O(d + d \log(n/d))$.

Define the following equivalence relations on the set of nodes of a graph. $u \sim v$ if and only if $\mathcal{V}(u) = \mathcal{V}(v)$, and $u \sim_t v$ if and only if $\mathcal{V}^t(u) = \mathcal{V}^t(v)$. Let Π be the partition of all nodes into equivalence classes of \sim , and Π_t the corresponding partition for \sim_t . It was proved in [32] (corollary 1) that all equivalence classes in Π are of equal size σ . In view of Proposition 2.2 this is also the case for Π_t , where $t \geq n - 1$. On the other hand, for smaller t , equivalence classes in Π_t may be of different sizes. Every equivalence class in Π_t is a union of some equivalence classes in $\Pi_{t'}$, for $t < t'$. The following result was proved in [25] (lemma 1). It says that if the sequence of partitions Π_t stops changing at some point, it will never change again.

Proposition 2.3 *If $\Pi_t = \Pi_{t+1}$, then $\Pi_t = \Pi$.*

For a set A , let $|A|$ denote its size. By definition of partitions Π_t , if $|\Pi_t| = |\Pi_{t+1}|$ then $\Pi_t = \Pi_{t+1}$. Hence in order to see when partitions stabilize, it is enough to check when their sizes stabilize. This allows us to modify Proposition 2.3 and obtain Proposition 2.4 below, which we will use in some of our algorithms.

Proposition 2.4 *If $|\Pi_t| = |\Pi_{t+1}|$, then $\Pi_t = \Pi$.*

Definition 2.2 (Level of symmetry) *For any graph G we define its level of symmetry λ as the smallest integer t , for which there exists a node v satisfying the condition $\{u : \mathcal{V}^t(u) = \mathcal{V}^t(v)\} = \{u : \mathcal{V}(u) = \mathcal{V}(v)\}$.*

By Proposition 2.1, for solvable graphs, the level of symmetry is the smallest t for which there is a node with a unique view at depth t . In general, the level of symmetry is the smallest integer t for which some equivalence class in Π_t has size σ .

Definition 2.3 *Define Λ to be the smallest integer t for which $\Pi_t = \Pi_{t+1}$.*

We have

Proposition 2.5 $\Lambda \leq D + \lambda$.

Proof: It is enough to show that if $\mathcal{V}^{D+\lambda}(w) = \mathcal{V}^{D+\lambda}(w')$, for some nodes w and w' , then $\mathcal{V}(w) = \mathcal{V}(w')$. Let v be a node for which $\{u : \mathcal{V}^\lambda(u) = \mathcal{V}^\lambda(v)\} = \{u : \mathcal{V}(u) = \mathcal{V}(v)\}$. Consider nodes w and w' for which $\mathcal{V}^{D+\lambda}(w) = \mathcal{V}^{D+\lambda}(w')$. Let p be a shortest path (coded as a sequence of ports) from w to v . Let v' be the node at the end of the same path p starting at w' . $\mathcal{V}^\lambda(v)$ is included in $\mathcal{V}^{D+\lambda}(w)$ and $\mathcal{V}^\lambda(v')$ is included in $\mathcal{V}^{D+\lambda}(w')$. It follows that $\mathcal{V}^\lambda(v) = \mathcal{V}^\lambda(v')$ and hence $\mathcal{V}(v) = \mathcal{V}(v')$. Suppose for contradiction that $\mathcal{V}(w) \neq \mathcal{V}(w')$. Let q and q' be paths in views $\mathcal{V}(w)$ and $\mathcal{V}(w')$ witnessing to their difference. Let \bar{p} be the reverse of path p . Then the concatenations $\bar{p}q$ and $\bar{p}q'$ are paths witnessing to the difference of $\mathcal{V}(v)$ and $\mathcal{V}(v')$, which gives a contradiction.

We fix a canonical linear order on all finite rooted trees with unlabeled nodes and labeled ports, e.g., as the lexicographic order of DFS traversals of these trees, starting from the root and exploring children of a node in increasing order of ports. For any subset of this class, the term “smallest” refers to this order. Since views at a depth t are such trees, we will elect as leader a node whose view at some depth is the smallest in some class of views. The difficulty is to establish when views at some depth are already unique for a solvable graph, and to decide fast if the graph is solvable, in the case of strong LE.

All our algorithms are written for a node u of the graph.

3 Weak leader election

In this section we show that the optimal time of weak leader election is $\Theta(D + \lambda)$, if either the diameter D of the graph or its size n is known to the nodes. We first give two algorithms, one working for the class of graphs of given diameter D and the other for the class of graphs of given size n , that elect a leader in time $O(D + \lambda)$ on graphs with diameter D and level of symmetry λ , whenever election is possible.

Our algorithms use the subroutine *COM* to exchange views at different depths with their neighbors. This subroutine is detailed in Algorithm 1.

Algorithm 2 works for the class of graphs of given diameter D .

Algorithm 1 $COM(i)$

let $\mathcal{V}^i(u)$ be the view of node u at depth i
send $\mathcal{V}^i(u)$ to all neighbors;
foreach neighbor v of u
 receive $\mathcal{V}^i(v)$ from v

Algorithm 2 $WLE\text{-known-diameter}(D)$

for $i := 0$ **to** $D - 1$ **do** $COM(i)$
compute $|\Pi_0|$; $j := 0$
repeat
 $COM(D + j)$; $j := j + 1$; compute $|\Pi_j|$
until $|\Pi_j| = |\Pi_{j-1}|$
 $V :=$ the set of nodes v in $\mathcal{V}^{D+j}(u)$ having the smallest $\mathcal{V}^{j-1}(v)$
elect as leader the node in V having the lexicographically smallest path from u

Theorem 3.1 *Algorithm 2 - $WLE\text{-known-diameter}(D)$ - elects a leader in every solvable graph of diameter D , in time $O(D + \lambda)$, where λ is the level of symmetry of the graph.*

Proof: All nodes of the graph find $|\Pi_0|$ (i.e., the number of different node degrees in the graph) after D rounds, and then they find sizes of consecutive partitions Π_j , for $j = 1, \dots, \Lambda + 1$ (by counting the number of distinct views at depth j). At this time the exit condition of the “repeat” loop is satisfied. All nodes stop simultaneously and elect a leader. Since the graph is solvable, by the definition of Λ and in view of Propositions 2.1 and 2.4, all elements of the partition Π_Λ are singletons (recall that equivalence classes in Π , and thus in Π_Λ , are of equal size σ , and $\sigma = 1$ in solvable graphs). Hence all nodes in V correspond to the same node in the graph and consequently all nodes elect as leader the same node. All nodes stop in round $D + \Lambda$, which is at most $2D + \lambda$ by Proposition 2.5.

Algorithm 3 works for the class of graphs of given size n .

Algorithm 3 $WLE\text{-known-size}(n)$

$i := 0$; $x := 1$;
while $x < n$ **do**
 $COM(i)$; $i := i + 1$
 for $j := 0$ **to** i **do**
 $L_j :=$ the set of nodes in $\mathcal{V}^i(u)$ at distance at most j from u (including u)
 $num_j :=$ the number of nodes in L_j with distinct views at depth $i - j$
 $x := \max \{num_j : j \in \{0, 1, \dots, i\}\}$
compute Λ and D
 $V :=$ the set of nodes v in $\mathcal{V}^i(u)$ having the smallest $\mathcal{V}^\Lambda(v)$
elect as leader the node in V having the lexicographically smallest path from u
while $i \leq D + \Lambda$ **do**
 $COM(i)$; $i := i + 1$ //Allow late nodes to terminate their protocol!

Theorem 3.2 *Algorithm 3 - $WLE\text{-known-size}(n)$ - elects a leader in every solvable graph of size n , in time $O(D + \lambda)$, where λ is the level of symmetry of the graph.*

Proof: Consider a node u . After $i \leq D + \Lambda$ rounds, node u gets a view $\mathcal{V}^i(u)$ that contains views $\mathcal{V}^\Lambda(v)$, for all nodes v . Since the graph is solvable, all views $\mathcal{V}^\Lambda(v)$ are different, and hence u exits the first “while” loop after seeing n different views at this depth. At this point u can reconstruct an isomorphic copy of the graph and hence compute D and Λ . The election rule is as in Algorithm 2. The second “while” loop

guarantees that every node v will be able to get $\mathcal{V}^{D+\Lambda}(v)$ and hence will exit the first while loop and elect the same leader.

In order to show that Algorithm 2 is optimal for the class of graphs of diameter D and Algorithm 3 is optimal for the class of graphs of size n we prove the following theorem. It shows a stronger property: both above algorithms have optimal complexity even among weak LE algorithms working only when all three parameters n , D and λ are known.

Theorem 3.3 *For any $D \geq 1$ and any $\lambda \geq 0$, with $(D, \lambda) \neq (1, 0)$, there exists an integer n and a solvable graph G of size n , diameter D and level of symmetry λ , such that every algorithm for weak LE working for the class of graphs of size n , diameter D and level of symmetry λ takes time at least $D + \lambda$ on the graph G ¹*

Before proving Theorem 3.3 we present a construction of a family Q_k of complete graphs (cliques) that will be used in the proof of our lower bound. The construction consists in assigning port numbers. In order to facilitate subsequent analysis we also assign labels to nodes of the constructed cliques. We will use induction to describe the construction of Q_k .

Q_1 is the single node graph. Q_2 is a 4-node clique with port numbers defined as follows. For a node u , we say that edge (i, j) is *incident* to u , if the edge corresponding to port i at node u corresponds to port j at some node v . Nodes in Q_2 are uniquely identifiable by the set of their incident edges. Below we assign distinct labels to the four nodes of the clique depending on the sets of their incident edges (see Fig. 2.a):

- set of edges $\{(0, 0), (1, 1), (2, 2)\}$ – corresponding to label a ;
- set of edges $\{(0, 0), (1, 1), (2, 0)\}$ – corresponding to label b ;
- set of edges $\{(0, 2), (1, 1), (2, 0)\}$ – corresponding to label c ;
- set of edges $\{(0, 2), (1, 1), (2, 2)\}$ – corresponding to label d .

We additionally assign colors 0, 1, and 2 to edges of Q_2 as follows: edges $\{a, b\}$ and $\{c, d\}$ get color 0, edges $\{a, c\}$ and $\{b, d\}$ get color 1, and edges $\{a, d\}$ and $\{b, c\}$ get color 2.

Q_3 is constructed starting from two disjoint copies of clique Q_2 as follows. Denote by Q_2 one of the copies and by \overline{Q}_2 the other one. For each node x in Q_2 , we denote by \overline{x} its corresponding node in \overline{Q}_2 .

Ports 3, 4, 5, and 6 are used to connect each node in Q_2 to all nodes in \overline{Q}_2 to construct an 8-node clique. We will use two types of edges. *Monochromatic edges* will have port 3 or 4 at both endpoints, while *skew edges* will have port 5 at one endpoint and port 6 at the other endpoint. For monochromatic edges we call *color* the port number at both their endpoints. We will also consider as monochromatic the edges in Q_2 .

More precisely, edges of color 3 connect nodes $\{a, \overline{c}\}$, $\{\overline{a}, c\}$, $\{b, \overline{d}\}$, and $\{\overline{b}, d\}$. Edges of color 4 connect nodes $\{a, \overline{d}\}$, $\{\overline{a}, d\}$, $\{b, \overline{c}\}$, and $\{\overline{b}, c\}$.

Notice that until now, this construction results in a graph for which each node has exactly one other node with the same view, e.g., a and \overline{a} have the same view. Uniqueness of all views at depth 2 is guaranteed in clique Q_3 by the definitions of the skew edges.

In particular, skew edges connect nodes:

- $\{a, \overline{a}\}$, $\{b, \overline{b}\}$, $\{c, \overline{c}\}$, and $\{d, \overline{d}\}$ with port 6 at nodes a, b, c , and d and port 5 at nodes $\overline{a}, \overline{b}, \overline{c}$, and \overline{d} ;
- $\{\overline{a}, b\}$, $\{\overline{b}, a\}$, $\{\overline{c}, d\}$, and $\{\overline{d}, c\}$ with port 6 at nodes $\overline{a}, \overline{b}, \overline{c}$, and \overline{d} and port 5 at nodes a, b, c , and d .

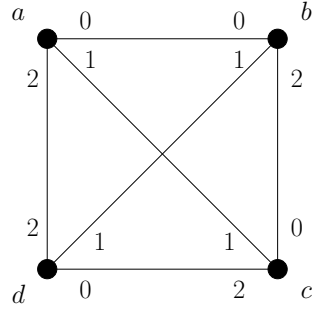
This concludes the construction of Q_3 (see Fig. 2.b).

A node x in Q_3 whose outgoing port 6 leads to a node y receives as its label the concatenation of the labels of nodes x and y in their respective copies of Q_2 (removing all overlines).

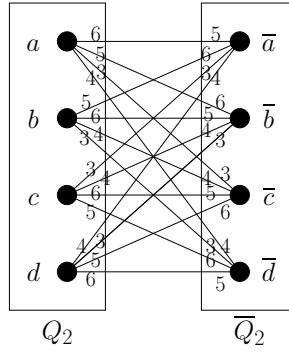
The complete labeling of nodes in Q_3 is: aa , ab , bb , ba , cc , cd , dd , and dc .

For $k \geq 3$, the clique Q_{k+1} is produced starting from disjoint copies Q_k and \overline{Q}_k of clique Q_k as follows.

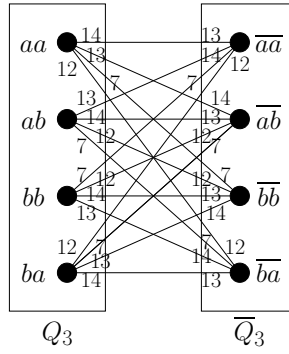
¹Notice that there is no solvable graph with $D = 1$ and $\lambda = 0$, because the latter condition, for solvable graphs, means that there is a node of a unique degree, contradicting the requirement $D = 1$, i.e., having a clique of at least 2 nodes.



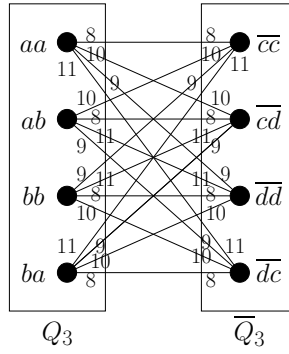
(a)



(b)



(c)



(d)

Figure 2: a) Clique Q_2 . b) Clique Q_3 (internal edges of Q_2 and \bar{Q}_2 are omitted). c) and d) Extracts of graph Q_4 .

The set of nodes of the clique Q_{k+1} is the union of the sets of nodes of Q_k and \overline{Q}_k . The skew edges connecting nodes of Q_k and \overline{Q}_k will have port numbers $(2^{k+1} - 2, 2^{k+1} - 3)$, while the monochromatic edges will use the same port i , for $i \in [2^k - 1, 2^{k+1} - 4]$, on both endpoints. We denote by α_j a string of length 2^j over the alphabet $\{a, b, c, d\}$. For $h \leq j$, α_h is the prefix of length 2^h of string α_j . We assign port numbers to the edges according to the following two rules.

1. **Skew edges:** the edge that has port $2^{k+1} - 2$ at a node $\alpha_{k-3}\beta_{k-3}$, has port $2^{k+1} - 3$ at its other endpoint $\alpha_{k-3}\beta_{k-3}$; the edge that has port $2^{k+1} - 2$ at a node $\alpha_{k-3}\beta_{k-3}$, has port $2^{k+1} - 3$ at its other endpoint $\alpha_{k-3}\gamma_{k-3}$ (where $\beta_{k-3} \neq \gamma_{k-3}$).
2. **Monochromatic edges:** let node $\alpha_{k-3}\beta_{k-3}$ and node $\overline{\gamma_{k-3}\delta_{k-3}}$ be connected by a monochromatic edge of color i ; then node $\overline{\alpha_{k-3}\beta_{k-3}}$ and node $\gamma_{k-3}\delta_{k-3}$ are connected by a monochromatic edge of color i . Moreover, for each $j \in [0, k - 3]$, any node $\alpha_j\eta$ is connected to some node $\gamma_j\zeta$ by a monochromatic edge of color i .

The label of a node v in Q_{k+1} is given by the concatenation of the labels of the node in Q_k (respectively in \overline{Q}_k) corresponding to v and its neighbor in \overline{Q}_k (respectively Q_k) connected by the skew edge $(2^{k+1} - 2, 2^{k+1} - 3)$ incident to v .

While rule 1 is constructive, rule 2 is not. However, a port assignment for monochromatic edges of Q_{k+1} that fulfills rule 2, can be obtained, for $k \geq 3$, by exploiting the already defined edges of Q_k as follows. If there is a monochromatic edge with color i between nodes $\{u, v\}$ in Q_k , then nodes $\{u, \bar{v}\}$ and nodes $\{\bar{u}, v\}$ are connected by a monochromatic edge with color $2^k + i - 1$. If $\{u, v\}$ are connected by the skew edge $(2^k - 3, 2^k - 2)$ incident to u , then nodes $\{u, \bar{v}\}$ and nodes $\{\bar{u}, v\}$ are connected by monochromatic edges of color $2^{k+1} - 4$. Now consider the skew edges from previous steps of the construction i.e., skew edges with port numbers $(2^j - 2, 2^j - 3)$ of graph Q_j , with $j \leq k - 1$. Let $\{u, \bar{u}, v, \bar{v}\}$ be four nodes connected by these edges, where $\{u, v\}$ are nodes from copy Q_{j-1} in the construction of Q_j , and nodes $\{\bar{u}, \bar{v}\}$ are the corresponding nodes in copy \overline{Q}_{j-1} . More precisely, let:

- edge $(2^j - 2, 2^j - 3)$ incident to u connect nodes u and \bar{u} ;
- edge $(2^j - 2, 2^j - 3)$ incident to v connect nodes v and \bar{v} ;
- edge $(2^j - 2, 2^j - 3)$ incident to \bar{u} connect nodes \bar{u} and v ;
- edge $(2^j - 2, 2^j - 3)$ incident to \bar{v} connect nodes \bar{v} and u .

Finally, let \bar{u} and \bar{v} from Q_j correspond to nodes w and z , respectively, in copy Q_k of the construction of Q_{k+1} . Then nodes $\{u, \bar{w}\}$, $\{\bar{u}, w\}$, $\{v, \bar{z}\}$, $\{\bar{v}, z\}$ are connected by monochromatic edges of color $2^k + 2^j - 4$; nodes $\{u, \bar{z}\}$, $\{\bar{u}, z\}$, $\{v, \bar{w}\}$, $\{\bar{v}, w\}$ are connected by monochromatic edges of color $2^k + 2^j - 3$. This concludes the definition of monochromatic edges of Q_{k+1} , and thus completes its construction. Extracts from graph Q_4 are depicted in Fig. 2.c and 2.d.

A node of the clique Q_k is said to be of *type* a, b, c , or d if it is obtained from a node with label a, b, c , or d (respectively) in a copy of Q_2 in the construction of Q_k . Notice that the type of a node corresponds to the first letter of its label. Consider a path p defined as a sequence of consecutive monochromatic and skew edges (i.e., p is uniquely defined by the sequence of outgoing port numbers). If a skew edge e is traversed from node u to v by p , then the *color* of e in p is given by its port number at u . We call p a *distinguishing* path for nodes x and y in Q_k , if it yields two different sequences of node types a, b, c , and d traversed proceeding along p , depending on whether the origin of p is x or y .

Lemma 3.1 *The clique Q_{k+1} has level of symmetry k .*

Proof: We will prove a stronger statement. We will show that, for $k \geq 2$, if $j \in [0, k - 2]$ and two nodes u and v in Q_{k+1} are such that their labels have a common prefix α_j but no common prefix α_{j+1} , then the views of u and v are identical up to depth $j + 1$ and differ at depth $j + 2$.

The base step of the inductive proof is for $k = 2$. Indeed, node aa in Q_3 has the same view as node ab at depth 1, while their views differ at depth 2. The same happens for the views of nodes bb and ba , cc and cd , dd and dc . The views of any other pair of nodes are different at depth 1.

Consider the construction of clique Q_{k+1} . Let $j \in [0, k-3]$ and consider two nodes u and v whose labels have a common prefix α_j but no common prefix α_{j+1} . Let $i \in [2^k - 1, 2^{k+1} - 2]$, and consider the neighbors u' and v' (of u and v respectively) that are the other endpoints of the edges having port i at u and v . The labels of u' and v' have a common prefix β_j . Indeed, if port i is a port number of a monochromatic edge, rule 2 immediately implies that u' and v' have a common prefix β_j . On the other hand, if i is a port number of a skew edge then the first half of the label of u' must coincide with that of u , and the first half of the label of v' with that of v in graph Q_k , which in turn implies that u' and v' have a common prefix $\beta_j = \alpha_j$ in graph Q_{k+1} . It follows that edges with colors in $[2^k - 1, 2^{k+1} - 2]$ cannot appear in any distinguishing path of minimal length between two nodes u and v whose labels have a common prefix of length up to 2^{k-3} , and therefore the depth at which views of such nodes differ in Q_{k+1} is the same as that of their corresponding nodes in Q_k .

It remains to consider the case of nodes $u \in Q_k$ and $\bar{u} \in \bar{Q}_k$, sharing a label prefix α_{k-2} in Q_{k+1} . For any such pair of nodes, if $i \in [2^k - 1, 2^{k+1} - 4]$ is the color of a monochromatic edge connecting u to \bar{v} , then another monochromatic edge of color i connects \bar{u} to v . Any pair of such nodes have the same view (at any depth) in Q_k . On the other hand, the skew edge $(2^{k+1} - 2, 2^{k+1} - 3)$ incident to u and that incident to \bar{u} lead, by rule 1, to two nodes having a common label prefix of length α_{k-3} , but no common prefix α_{k-2} (the same reasoning applies to the skew edges $(2^{k+1} - 3, 2^{k+1} - 2)$ incident to \bar{u} and u). These nodes have identical views up to depth $k - 2$ in Q_k , but distinct at depth $k - 1$, by the inductive hypothesis. Hence u and \bar{u} have the same view up to depth $k - 1$ and distinct views at depth k in Q_{k+1} , which concludes the proof.

Remark 3.1 *After the publication of the conference version of this paper a more general version of Lemma 3.1 has been proved in [10]: for any $D \leq n$ there exists a $\Theta(n)$ -node graph of diameter $\Theta(D)$ with level of symmetry $\Omega(D \log(n/D))$.*

We will also use the following family of cliques \tilde{Q}_k . \tilde{Q}_1 is the clique on 2 nodes, with port number 0. \tilde{Q}_2 is a clique on 4 nodes, where all nodes have the same set of incident edges $\{(0, 0), (1, 1), (2, 2)\}$. For $k \geq 2$, \tilde{Q}_{k+1} is a clique obtained from two disjoint copies of Q_k . The construction of \tilde{Q}_{k+1} mimics the construction of Q_{k+1} for all edges but the skew edges between nodes in Q_k and \bar{Q}_k , that are replaced by monochromatic edges with port number $2^{k+1} - 2$ at both endpoints, connecting nodes u and \bar{u} , and by monochromatic edges with port number $2^{k+1} - 3$ at both endpoints, connecting nodes $\alpha_{k-3}\alpha_{k-3}$ and $\alpha_{k-3}\beta_{k-3}$ and nodes $\bar{\alpha}_{k-3}\alpha_{k-3}$ and $\alpha_{k-3}\beta_{k-3}$. Notice that, in graph \tilde{Q}_k , nodes x and \bar{x} have identical views. Nevertheless, in order to describe our construction, we artificially assign to nodes x and \bar{x} the label they would respectively receive in the construction of graph Q_k .

We finally define a family of graphs that allow us to prove Theorem 3.3. For any pair of integers (D, λ) , with $D \geq 2$ and $\lambda \geq 2$, the graph $R_{D,\lambda}$ is obtained using one copy of graph $Q_{\lambda+1}$ and $2D - 1$ copies of graph $\tilde{Q}_{\lambda+1}$. The construction of graph $R_{D,\lambda}$ proceeds as follows. Arrange $2D - 1$ disjoint copies of $\tilde{Q}_{\lambda+1}$ and one copy of $Q_{\lambda+1}$ in a cyclic order. Connect each node in a clique with all nodes in the subsequent clique. Let $\{x, y\}$ be two nodes in $Q_{\lambda+1}$ and let i be the color that would be assigned to edge (x, \bar{y}) in the construction of $\tilde{Q}_{\lambda+2}$. Assign port numbers $(i, i + 2^{\lambda+1})$ to the edge connecting node x' in some clique to node y'' in the subsequent clique, where x' has label x and y'' has label y .

A distinguishing path in $R_{D,\lambda}$ is defined in the same way as in $Q_{\lambda+1}$, which is possible, since in both graphs each node has type a , b , c , or d .

Lemma 3.2 *Let $\lambda \geq 2$ and $D \geq 2$. Let x and y be two nodes in $Q_{\lambda+1}$ and let $\ell \leq \lambda - 1$ be the maximum depth at which views of x and y in $Q_{\lambda+1}$ are identical. Then the views at depth ℓ of nodes x and y belonging to the copy of $Q_{\lambda+1}$ in $R_{D,\lambda}$ are identical.*

Proof: It follows from the definition that the length of a shortest distinguishing path for nodes x and y in $Q_{\lambda+1}$ is ℓ . By construction of $Q_{\lambda+1}$, nodes x and y have labels with an identical prefix of length

$2^{\ell+1}$. Suppose, for contradiction, that views of x and y at depth ℓ are different in $R_{D,\lambda}$. Hence a shortest distinguishing path for these nodes has length $t < \ell$ in $R_{D,\lambda}$. Let p be a distinguishing path of length t in $R_{D,\lambda}$ for nodes x and y . We will show how to construct a distinguishing path p' of length at most t in $Q_{\lambda+1}$, which will give a contradiction.

Edge colors $\{0, 1, 2\}$ inside each clique $\tilde{Q}_{\lambda+1}$ are assigned according to the rules used for cliques $Q_{\lambda+1}$. Monochromatic edges with port number i at both endpoints get color i . For edges connecting nodes from different cliques in $R_{D,\lambda}$ we assign as color the smaller of their port numbers. (Notice that edges connecting nodes with the same label in their cliques will get the same color $2^{\lambda+2} - 2$ but these edges will be subsequently deleted in the construction of the distinguishing path.) For the remaining edges (i.e., skew edges in the construction of a graph Q_k , for some $k \leq \lambda$), the color is defined by the outgoing port number, according to path p .

The distinguishing path p' is constructed as follows. Consider all edges e in p , in reverse order. If e has color $h = 2^{\lambda+2} - 2$ then we remove it; if e has color h such that $2^{\lambda+2} - 3 \geq h \geq 2^{\lambda+1} - 1$ then we replace it with an edge with color $h - 2^{\lambda+1}$. This corresponds to replacing edges going from one copy of $\tilde{Q}_{\lambda+1}$ or from $Q_{\lambda+1}$ to another copy of $\tilde{Q}_{\lambda+1}$ or to Q_λ with internal edges of some clique $\tilde{Q}_{\lambda+1}$ or of $Q_{\lambda+1}$. Deleted edges are those going to corresponding nodes of different cliques, hence their deletion corresponds to removing self loops. We show that path p' is distinguishing for nodes x and y in $Q_{\lambda+1}$.

Indeed, the only edge replacements that could modify the sequence of a , b , c , and d types yielded by paths p and p' , when starting from nodes x and y , are those of edges with color $2^{\lambda+2} - 3$, as these edges may be replaced by skew edges of the last step of the construction of the clique $Q_{\lambda+1}$, that are defined differently in cliques $\tilde{Q}_{\lambda+1}$. For each such edge leading to a node with label $\alpha_{\lambda-2}\beta_{\lambda-2}$, the corresponding skew edge in $Q_{\lambda+1}$ leads to a node with label $\alpha_{\lambda-2}\gamma_{\lambda-2}$. Hence, views in $Q_{\lambda+1}$ of nodes $\alpha_{\lambda-2}\beta_{\lambda-2}$ and $\alpha_{\lambda-2}\gamma_{\lambda-2}$ are identical up to depth $\lambda - 1$, as shown in the proof of Lemma 3.1, and thus the path p' , of length at most t , is distinguishing for x and y in $Q_{\lambda+1}$, contradiction.

Proof of Theorem 3.3: Lemma 3.3 below proves the theorem if either D or λ are less than 2. Here we give the general argument for $D, \lambda \geq 2$. Consider the clique $\tilde{Q}_{\lambda+1}$ antipodal to the clique $Q_{\lambda+1}$ in graph $R_{D,\lambda}$. Consider nodes x and \bar{x} from this clique. Any distinguishing path for nodes x and \bar{x} in $R_{D,\lambda}$ must contain a node from $Q_{\lambda+1}$. Let q be a minimum length distinguishing path for nodes x and \bar{x} and assume without loss of generality that y and \bar{y} are the first nodes from $Q_{\lambda+1}$ found along path q , if starting from x and \bar{x} , respectively. By Lemmas 3.1 and 3.2, nodes y and \bar{y} have the same views at depth $\lambda - 1$ and different views at depth λ in the graph $R_{D,\lambda}$. Thus the minimum length distinguishing path in $R_{D,\lambda}$ for y and \bar{y} has length $\lambda - 1$. Since nodes y and \bar{y} are at distance D from x and \bar{x} , respectively, the views at depth $D + \lambda - 1$ of x and \bar{x} are identical.

The following lemma proves Theorem 3.3 in the case when either D or λ are small, thus concluding the proof of Theorem 3.3.

Lemma 3.3 *For $D = 1$ and any $\lambda \geq 1$, and for any $D \geq 2$ and $0 \leq \lambda \leq 1$, there exists an integer n and a solvable graph G of size n , diameter D and level of symmetry λ , such that every algorithm for weak LE, working for the class of graphs of size n , diameter D and level of symmetry λ , takes time at least $D + \lambda$ on the graph G .*

Proof: Consider the following cases.

Case 1. $\lambda = 0$ and $D \geq 2$.

The required graph G is depicted in Fig. 3. The lower bound D on the time of weak LE is straightforward.

Case 2. $\lambda = 1$ and $D \geq 2$.

The construction of the graph G is analogous to the general construction of $R_{D,\lambda}$. We take one copy of clique Q_2 and $2D - 1$ copies of clique \tilde{Q}_2 in circular order. In copies of cliques \tilde{Q}_2 we arbitrarily assign labels a , b , c , and d to nodes. The argument remains the same as in the general case.

Case 3. $\lambda \geq 1$ and $D = 1$.

Consider the clique constructed from cliques $Q_{\lambda+1}$ and $\tilde{Q}_{\lambda+1}$ by connecting each node with label x in $Q_{\lambda+1}$ to each node with label y in $\tilde{Q}_{\lambda+1}$ by an edge having the color that would be assigned to edge (x, \bar{y}) in the construction of $\tilde{Q}_{\lambda+2}$. The obtained clique requires time $1 + \lambda$ for weak LE.

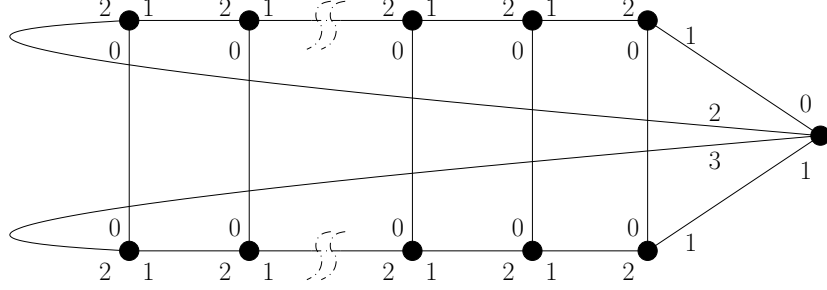


Figure 3: Graph G of diameter $D \geq 1$ and level of symmetry 0.

4 Strong leader election

For strong leader election more knowledge is required to accomplish it, and even more knowledge is needed to perform it fast. We first prove that knowledge of the diameter D is not sufficient for this task. The idea is to show, for sufficiently large D , one solvable and one non-solvable graph of diameter D , such that both graphs have the same sets of views.

Theorem 4.1 *For any $D \geq 4$ there is no strong LE algorithm working for all graphs of diameter D .*

Proof: Let $k \geq 3$. We will use the following family of graphs T_k (see also Fig. 4.a). Consider a $2 \times 2k$ torus τ . Let R and R' be the two rings of size $2k$ in τ . For each node $u \in R$, let u' be its unique neighbor from R' in τ . The graph T_k is obtained by connecting each node $u \in R$ with nodes v' and w' in R' , where v and w are the two neighbors of u in R . An additional node z of degree 2, connected to a pair of nodes u, u' , completes the construction of T_k . The assignment of port numbers can be performed arbitrarily and is thus omitted in the construction.

Consider the following family M_k of graphs (see also Fig. 4.b). The graph M_k is obtained from two copies of the graph T_k . Let us call T_k one copy, and \bar{T}_k , the other. Similarly we call \bar{u} the copy in \bar{T}_k of node $u \in T_k$. The graph M_k is obtained by removing all edges $\{u, v'\}$ and $\{\bar{u}, \bar{v}'\}$ from T_k and \bar{T}_k , and replacing them with edges $\{u, \bar{v}'\}$ and $\{\bar{u}, v'\}$, maintaining the same port numbers.

Since the graph T_k has exactly one node of degree 2, it is solvable for any assignment of port numbers. By the construction of graph M_k , each node u from one copy of T_k has the same view as the corresponding node \bar{u} from the other copy. It follows that graph M_k is not solvable. To prove the theorem, it is thus enough to show that graphs T_k and M_k have the same diameter. Indeed, this will imply that no algorithm knowing only the diameter can tell apart some solvable graph from an unsolvable one.

The diameter of graph T_k is $k + 1$. Indeed, $k + 1$ is the distance from the unique node z of degree two to the two nodes in rings R and R' that are antipodal to the neighbors of z . All other pairs of nodes are at distance at most k . Consider the graph M_k . Since no edge from the original torus in the copies of T_k is modified in the construction of M_k , we have that all pairs of nodes from the copy T_k , as well as all pairs from the copy \bar{T}_k are within distance $k + 1$ from each other. By using one edge $\{u, \bar{v}'\}$ each node $u \in R$ can reach any node \bar{v}' in \bar{R}' within k steps (similarly for $u' \in R'$). Nodes $u \in R$ and $\bar{u} \in \bar{R}$ are at distance 3 in M_k , for any $k \geq 3$. The two copies of the node of degree two are at distance 4 in M_k , for any $k \geq 3$. Hence, for any $k \geq 3$, all pairs of nodes in M_k are within distance $k + 1$ from each other. On the other hand, the distance in M_k from a node u in R and a node \bar{x} in \bar{R} , where x is the antipodal node of u in R , is equal to $k + 1$. Hence the diameter of M_k is $k + 1$, which concludes the proof.

By contrast, knowledge of the size n alone is enough to accomplish strong leader election, but (unlike for weak LE), it may be slow. We will show that optimal time for strong LE is $\Theta(n)$ in this case. We first show Algorithm 4, working in time $O(n)$.

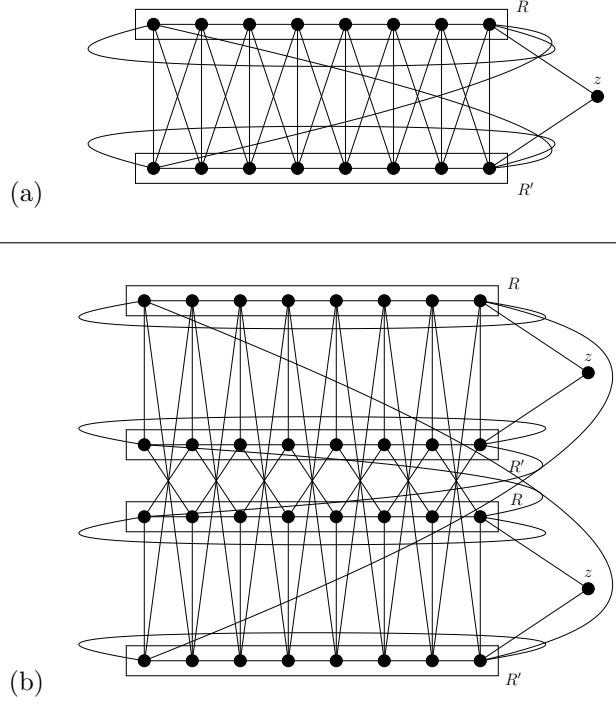


Figure 4: a) The graph T_4 . b) The graph M_4 .

Algorithm 4 SLE-known-size(n)

for $i := 0$ **to** $2n - 3$ **do** $COM(i)$

$L :=$ the set of nodes in $\mathcal{V}^{2n-2}(u)$ at distance at most $n - 1$ from u (including u)

$num :=$ the number of nodes in L with distinct views at depth $n - 1$

if $num < n$ **then** report “LE impossible”

else

$V :=$ the set of nodes v in $\mathcal{V}^{2n-2}(u)$ having the smallest $\mathcal{V}^{n-1}(v)$

elect as leader the node in V having the lexicographically smallest path from u

Theorem 4.2 *Algorithm 4 - SLE-known-size(n) - performs strong LE in the class of graphs of size n , in time $O(n)$.*

Proof: After $2n - 2$ rounds of communication every node u has view $\mathcal{V}^{2n-2}(u)$, which contains views $\mathcal{V}^{n-1}(v)$ of all nodes v . A copy of each node appears in the set L . By Propositions 2.1 and 2.2 the graph is solvable if and only if there are exactly n different views $\mathcal{V}^{n-1}(v)$ of nodes in L . Hence the report “LE impossible” is correct when $\text{num} < n$. If LE is possible, the election rule is the same as in the weak LE algorithms. Hence all nodes elect the same leader in this case.

Our next result shows that Algorithm 4 is optimal if only n is known. Compared to Theorem 3.2 it shows that the optimal time of strong LE with known size can be exponentially slower than that of weak LE with known size. Indeed, it shows that strong LE may take time $\Omega(n)$ on some graphs of diameter logarithmic in their size and having level of symmetry 0, while weak LE takes time $O(\log n)$, on any solvable graph of diameter $O(\log n)$ and level of symmetry 0.

The high-level idea of proving that Algorithm 4 is optimal if only n is known is the following. For arbitrarily large n , we show one solvable and one non-solvable graph of size n , such that there are nodes in one graph having the same view at depth $\Omega(n)$ as some nodes of the other.

Theorem 4.3 *For arbitrarily large n there exist graphs H_n of size n , level of symmetry 0 and diameter $O(\log n)$, such that every strong LE algorithm working for the class of graphs of size n takes time $\Omega(n)$ on graph H_n .*

Proof: Consider the following family of graphs G_k , for $k \geq 2$. The construction of graph G_k starts from a ring with $(2^k \cdot 5) - 4$ nodes, where all edges have port numbers 0 and 1 at the endpoints. All nodes in G_k will have degree 3 at the end of the construction. Hence all edges that do not belong to the original ring will have port number 2 at both endpoints.

Pick an edge $\{u, v\}$ of the ring and call u its first node. Consider the whole ring as a segment S_k of $(2^k \cdot 5) - 3$ edges having edge $\{u, v\}$ as its first and last edge (hence nodes u and v appear twice in S_k).

Let $\{x, y\}$ be the central edge of segment S_k , where x precedes y in S_k . Connect the second endpoint of the first edge of S_k (i.e., node v) to x . Connect the first endpoint of the last edge of S_k (i.e., node u) to y . Let S'_{k-1} be the segment obtained from S_k by removing the first edge, edge $\{x, y\}$ and all subsequent edges. Let S''_{k-1} be the segment obtained from S_k by removing all edges preceding edge $\{x, y\}$ (included) and the last edge. If S'_{k-1} and S''_{k-1} have length larger than 2, proceed recursively on both segments. If S'_{k-1} and S''_{k-1} have length 2, connect their central nodes. This concludes the construction of graph G_k (see Fig. 5.a for an example). For $n = (2^k \cdot 5) - 4$ we define $H_n = G_k$.

Since the length of a segment S_i is given by $|S'_{i-1}| + |S''_{i-1}| + 3 = 2|S_{i-1}| + 3$ and $|S_0| = 2$, we get the formula $|S_i| = (2^i \cdot 5) - 3$. Hence every segment S_i has a central edge if $|S_i| > 2$. The diameter of graph G_k is at most $4k + 2$. Indeed, starting from node u it is possible to reach any other node in G_k using at most $2k + 1$ edges. Views of all nodes in G_k are identical, hence leader election in graph G_k is impossible.

Consider the family of graphs G'_k , obtained, for any $k \geq 2$, from a ring with $(2^k \cdot 5) - 5$ nodes, where all edges have port numbers 0 and 1 at the endpoints, as follows. Add a new node v and connect it to some node u in the ring; divide the remaining $(2^k \cdot 5) - 6$ nodes of the ring into $\lfloor ((2^k \cdot 5) - 6)/4 \rfloor$ groups of 4 consecutive nodes and connect the first node of each group with the third one of the same group, as well as the second with the fourth. Again, new edges joining nodes of the original ring have port number 2 at both endpoints. Leave the remaining 2 consecutive nodes unchanged (see Fig. 5.b for an example).

Suppose, for contradiction, that some algorithm A correctly solves strong LE for all graphs of known size $n = (2^k \cdot 5) - 4$, where $k = 2, 3, \dots$, in time at most $n/5$. In particular, for any $k \geq 5$, algorithm A must report “LE impossible” in graph G_k in time at most $2^k - 1$. However, any node w at distance at least $2^k + 3$ from the unique node of degree 1 in graph G'_k has the same view, up to depth $2^k - 1$, as nodes in G_k . Thus node w would report “LE impossible” in graph G'_k and algorithm A would fail to elect a leader in the solvable graph G'_k , contradicting the assumption that A solves strong LE for all graphs of known size n , for $n = (2^k \cdot 5) - 4$. This contradiction implies that every algorithm solving strong LE with known size n must take time $\Omega(n)$ on graphs H_n which have diameter $O(\log n)$ and level of symmetry 0.

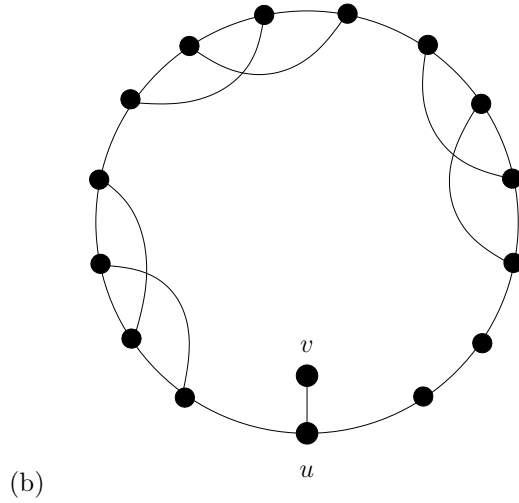
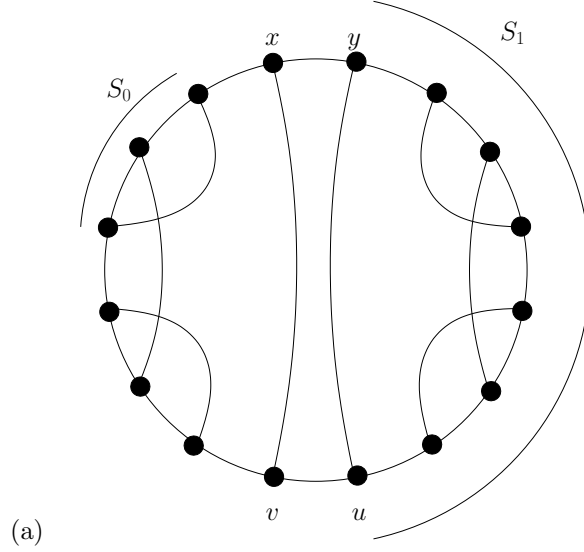


Figure 5: a) Graph G_2 . b) Graph G'_2 .

Algorithm 5 SLE-known-size-and-diameter(n, D)

for $i := 0$ **to** $D - 1$ **do** $COM(i)$

compute $|\Pi_0|$; $j := 0$

repeat

$COM(D + j)$; $j := j + 1$; construct Π_j

until $|\Pi_j| = |\Pi_{j-1}|$

if $|\Pi_j| < n$ **then** report “LE impossible”

else

$V :=$ the set of nodes v in $\mathcal{V}^{D+j}(u)$ having the smallest $\mathcal{V}^{j-1}(v)$

elect as leader the node in V having the lexicographically smallest path from u

We finally show that if both D and n are known, then the optimal time of strong LE is $\Theta(D + \lambda)$, for graphs with level of symmetry λ . The upper bound is given by Algorithm 5 - SLE-known-size-and-diameter(n, D). The algorithm is a variation of Algorithm 2 - WLE-known-diameter(D) - with an added test on the size of the partition Π_j after exiting the “repeat” loop.

The following result says that Algorithm SLE-known-size-and-diameter(n, D) is fast. In fact, compared to Theorems 4.1 and 4.3, it shows that while knowledge of the diameter alone does not help to accomplish strong LE, when this knowledge is added to the knowledge of the size, it may exponentially decrease the time of strong LE.

Theorem 4.4 *Algorithm SLE-known-size-and-diameter(n, D) performs strong LE in the class of graphs of size n and diameter D , in time $O(D + \lambda)$, for graphs with level of symmetry λ .*

Proof: The time complexity $O(D + \lambda)$ of Algorithm SLE-known-size-and-diameter(n, D) on solvable graphs follows directly from the analysis of Algorithm 2. The test on the number of sets in partition Π_j correctly identifies solvable and non-solvable graphs. Indeed, in view of Proposition 2.3, $\Pi_j = \Pi$ and it is a direct consequence of Proposition 2.1 that leader election is possible on an n -node graph, if and only if, Π contains n sets. Proposition 2.5 guarantees completion time $O(D + \lambda)$ on non-solvable graphs, while uniqueness of the elected leader on solvable graphs follows from the analysis of Algorithm 2. This concludes the proof.

Since the lower bound in Theorem 3.3 was formulated for known n , D and λ , it implies a matching lower bound for the optimal time of strong LE with known n and D , showing that this time is indeed $\Theta(D + \lambda)$ for graphs with level of symmetry λ .

5 Conclusion

We established the optimal time of weak and strong leader election, depending on the knowledge of the size n and of the diameter D of the graph. For each scenario the upper bounds were shown by proposing an algorithm, and matching lower bounds were proved. The optimal time turned out to depend on the knowledge of one or both of the parameters n and D .

Notice that the comparison of assumptions for our matching upper and lower bounds in various scenarios of weak and strong leader election shows that, while the level of symmetry may significantly influence optimal election time, the *knowledge* of this level is not important. Indeed, for each task and for various combinations of knowledge of the size n and/or the diameter D , adding the knowledge of the symmetry level λ does not help. More precisely, weak LE is impossible knowing neither n nor D , even if knowledge of λ , e.g., $\lambda = 0$ is added, cf. [32]. On the other hand, Theorems 3.1, 3.2, and 3.3 show that with known D or n , optimal time of weak LE is $\Theta(D + \lambda)$, regardless of whether λ is known or not (but the time itself depends on both D and λ). The same is true for strong LE. Indeed, the impossibility result in Theorem 4.1 and the lower bound from Theorem 4.3 hold even for known $\lambda = 0$, while the upper bound from Theorem 4.2 does not require the knowledge of λ . For known D and n , the upper bound $O(D + \lambda)$ holds without knowing λ by Theorem 4.4, while the lower bound carried over from Theorem 3.3 holds even when λ is known.

It would be interesting to investigate other complexity measures of the leader election problem, such as the bit or message complexity of communication needed to accomplish it, in relation to the knowledge of the three parameters of the network used in our present study.

References

- [1] D. Angluin, Local and Global Properties in Networks of Processors (Extended Abstract), *Proc. of the 12th Annual ACM Symposium on Theory of Computing (STOC 1980)*, 82–93
- [2] H. Attiya, M. Snir, and M. Warmuth, Computing on an Anonymous Ring, *Journal of the ACM* 35, (1988), 845–875.

- [3] H. Attiya and M. Snir, Better Computing on the Anonymous Ring, *Journal of Algorithms* 12, (1991), 204–238.
- [4] P. Boldi, S. Shammah, S. Vigna, B. Codenotti, P. Gemmell, and J. Simon, Symmetry Breaking in Anonymous Networks: Characterizations. *Proc. 4th Israel Symposium on Theory of Computing and Systems, (ISTCS 1996)*, 16–26.
- [5] P. Boldi and S. Vigna, Computing Anonymously With Arbitrary Knowledge. *Proc. 18th ACM Symp. on Principles of Distributed Computing*, 1999, 181–188.
- [6] J.E. Burns, A formal Model for Message Passing Systems, *Tech. Report TR-91*, Computer Science Department, Indiana University, Bloomington, September 1980.
- [7] J. Chalopin, Local Computations on Closed Unlabelled Edges: The Election Problem and the Naming Problem. *Proc. SOFSEM 2005: Theory and Practice of Computer Science*, 82–91.
- [8] J. Chalopin, A.W. Mazurkiewicz, and Y. Métivier, Labelled (Hyper)Graphs, Negotiations and the Naming Problem. *Proc. 4th international conference on Graph Transformations (ICGT 2008)*, 54–68.
- [9] J. Chalopin and Y. Métivier, Election and Local Computations on Edges. *Proc. Foundations of Software Science and Computation Structures (FoSSaCS 2004)*, 90–104.
- [10] D. Dereniowski, A. Kosowski, and D. Pajak, Distinguishing Views in Symmetric Networks: A Tight Lower Bound, *Tech. Report hal-00875370*, version 2, Inria, October 2013.
- [11] K. Diks, E. Kranakis A. Malinowski, and A. Pelc, Anonymous Wireless Rings, *Theoretical Computer Science* 145 (1995), 95–109.
- [12] S. Dobrev and A. Pelc, Leader Election in Rings with Nonunique Labels, *Fundamenta Informaticae* 59 (2004), 333–347.
- [13] P. Flocchini, E. Kranakis, D. Krizanc, F.L. Luccio, and N. Santoro, Sorting and Election in Anonymous Asynchronous Rings, *J. Parallel Distrib. Comput.* 64(2), (2004), 254–265.
- [14] G.N. Fredrickson and N.A. Lynch, Electing a Leader in a Synchronous Ring, *Journal of the ACM* 34 (1987), 98–115.
- [15] E. Fusco, A. Pelc, How Much Memory is Needed for Leader Election, *Distributed Computing* 24 (2011), 65–78.
- [16] M.A. Haddar, A.H. Kacem, Y. Métivier, M. Mosbah, and M. Jmaiel, Electing a Leader in the Local Computation Model Using Mobile Agents. *Proc. 6th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 2008)*, 473–480.
- [17] J. M. Hendrickx, Views in a Graph: To Which Depth Must Equality be Checked?, *IEEE Transactions on Parallel and Distributed Systems*, 25 (2014), 1907–1912.
- [18] D.S. Hirschberg, and J.B. Sinclair, Decentralized Extrema-Finding in Circular Configurations of Processes, *Communications of the ACM* 23 (1980), 627–628.
- [19] T. Jurdzinski, M. Kutylowski, and J. Zatoptionski, Efficient Algorithms for Leader Election in Radio Networks. *Proc. 21st ACM Symp. on Principles of Distributed Computing (PODC 2002)*, 51–57.
- [20] D. Kowalski, and A. Pelc, Leader Election in Ad Hoc Radio Networks: A Keen Ear Helps. *Journal of Computer and System Sciences* 79 (2013): 1164–1180.
- [21] E. Kranakis, Symmetry and Computability in Anonymous Networks: A Brief Survey. *Proc. 3rd Int. Conf. on Structural Information and Communication Complexity*, 1997, 1–16.

- [22] E. Kranakis, D. Krizanc, and J. van der Berg, Computing Boolean Functions on Anonymous Networks, *Information and Computation* 114, (1994), 214–236.
- [23] G. Le Lann, Distributed Systems - Towards a Formal Approach. *Proc. IFIP Congress*, 1977, 155–160, North Holland.
- [24] K. Nakano and S. Olariu, Uniform Leader Election Protocols for Radio Networks, *IEEE Transactions on Parallel and Distributed Systems* 13 (2002), 516–526.
- [25] N. Norris, Universal Covers of Graphs: Isomorphism to Depth $N - 1$ Implies Isomorphism to All Depths, *Discrete Applied Mathematics* 56, 1 (1995), 61–74.
- [26] D. Peleg, Distributed Computing, a Locality-Sensitive Approach, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia 2000.
- [27] G.L. Peterson, An $O(n \log n)$ Unidirectional Distributed Algorithm for the Circular Extrema Problem, *ACM Transactions on Programming Languages and Systems* 4 (1982), 758–762.
- [28] N. Sakamoto, Comparison of Initial Conditions for Distributed Algorithms on Anonymous Networks. *Proc. 18th ACM Symp. on Principles of Distributed Computing (PODC 1999)*, 173–179.
- [29] D.E. Willard, Log-Logarithmic Selection Resolution Protocols in a Multiple Access Channel, *SIAM J. on Computing* 15 (1986), 468–477.
- [30] M. Yamashita and T. Kameda, Computing on Anonymous Networks. *Proc. 7th ACM Symp. on Principles of Distributed Computing (PODC 1988)*, 117–130.
- [31] M. Yamashita and T. Kameda, Electing a Leader when Processor Identity Numbers are not Distinct. *Proc. 3rd Workshop on Distributed Algorithms*, LNCS Vol 392, Springer-Verlag, (WDAG 1989), 303–314.
- [32] M. Yamashita and T. Kameda, Computing on Anonymous Networks: Part I - Characterizing the Solvable Cases, *IEEE Trans. Parallel and Distributed Systems* 7 (1996), 69–89.